Mars Climate Orbiter

- Purpose: to relay signals from the Mars Polar Lander once it reached the surface of the planet
- Disaster: smashed into the planet instead of reaching a safe orbit
- Why: Software bug - failure to convert English measures to metric values
- $165M

Mars Climate Orbiter

Shooting Down of Airbus 320

- 1988
- US Vicennes shot down Airbus 320
- Mistook Airbus 320 for a F-14
- 290 people dead
- Why: Software bug - cryptic and misleading output displayed by the tracking software

THERAC-25 Radiation Therapy

- THERAC-25, a computer-controlled radiation-therapy machine
- 1986: two cancer patients at the East Texas Cancer Center in Tyler received fatal radiation overdoses
- Why: Software bug - mishandled race condition (i.e., miscoordination between concurrent tasks)

London Ambulance Service

- London Ambulance Service Computer Aided Dispatch (LASCAD)
- Purpose: automate many of the human-intensive processes of manual dispatch systems associated with ambulance services in the UK
  - functions: Call taking
- Failure of the London Ambulance Service on 26 and 27 November 1992

"Nice of You to Turn Up"

- Load increased
- Emergencies accumulated
- System made incorrect allocations
  - more than one ambulance being sent to the same incident
  - the closest vehicle was not chosen for the emergency
- At 23:00 on October 28 the LAS eventually instigated a backup procedure, after the death of at least 20 patients
More...

  - Errors in medical software have caused deaths

More...

  - British destroyer H.M.S. Sheffield; sunk in the Falkland Islands war; ship's radar warning system software allowed missile to reach its target
  - An Air New Zealand airliner crashed into an Antarctic mountain
  - North American Aerospace Defense Command reported that the U.S. was under missile attack; traced to faulty computer software - generated incorrect signals
  - Manned space capsule Gemini V missed its landing point by 100 miles; software ignored the motion of the earth around the sun

More...

  - An error in an aircraft design program contributed to several serious air crashes
  - Dallas/Fort Worth air-traffic system began spitting out gibberish in the Fall of 1989 and controllers had to track planes on paper

More...

- Software Reliability: Principles & Practice, p. 25, by G. J. Myers
  - Apollo 8 spacecraft erased part of the computer's memory
  - Eighteen errors were detected during the 10-day flight of Apollo 14
  - An error in a single FORTRAN statement resulted in the loss of the first American probe to Venus

More...

- An Airbus A320 crashes at an air show
- A China Airlines Airbus Industries A300 crashes on April 26, 1994 killing 264
- Ariane 5 satellite launcher malfunction was caused by a faulty software exception routine resulting from a bad 64-bit floating point to 16-bit integer conversion

More...

- ACM SIGSOFT Software Engineering Notes, vol. 6, no. 2
  - F-18 fighter plane crashed due to a missing exception condition
- ACM SIGSOFT Software Engineering Notes, vol. 9, no. 5
  - F-14 fighter plane was lost to uncontrollable spin, traced to tactical software
More...

- CyberSitter censors "menu */ #define" because of the string "nu...de"
- London’s Docklands Light Railway - train stopped in the middle of nowhere due to future station location programmed in software
- ACM SIGSOFT Software Engineering Notes, vol. 12, no. 3
  - Chicago cat owners were billed $5 for unlicensed dachshunds. A database search on "DHC" (for dachshunds) found "domestic house cats' with shots but no license

More...

- and many many more ....

Russia: Software bug made Soyuz stray

STAVROPOL CITY, Russia (AP) — A computer software error likely sent a Russian spacecraft into a non-ballistic descent that subjected the three crew on board to destabilizing gravity loads that made it hard to breathe, space experts said Tuesday.

"For me, for a moment, it felt like I was Atkins and I had the weight of the whole world on my shoulders," astronaut Donald Pettit, with a little wince, told reporters at a crowded news conference.

Economic Impact

- NIST study
  - On CNN.com - April 27, 2003

Expectation

- Can’t we expect software to execute correctly?
- Carefully made programs
  - 5 faults/1000 LOC
  - 1M LOC will have 5000 faults
- Windows XP has 45M LOC
  - How many faults?
  - 45 x 5000 = 225,000
  - Why not remove the faults?
Joke?

- "If the automobile industry had developed like the software industry, we would all be driving $25 cars that get 1,000 miles to the gallon."
- "Yeah, and if cars were like software, they would crash twice a day for no reason, and when you called for service, they'd tell you to reinstall the engine."

How Cars Are Engineered
(A Simple View)

- User requirements
  - Engine power, all-wheel, seating, comfort, MP3 player!
- Detailed design
  - Blueprints, design documents
- Verify design
  - Simulation, prototyping
- Develop parts (components)
  - Test each component
  - Components may be reused
  - Mass produced
- Assemble the car
  - Test the car (Front/side crash tests, Stability tests)
  - Usability testing (Feedback from drivers/passengers)

How Cars Are REALLY Engineered (A Detailed View)

But Seriously

- Features of many LEGO parts
  - Modularity
  - Reusability
    - Each part can be used in different places and ways
  - Flexibility of design
  - Compatibility
    - With other LEGO sets
  - Building-blocks
Detailed Design and Specifications

- Galvanized Bridge Wire for Parallel Wire Bridge Cable: Recommended diameter .196 inch.
- Galvanized Bridge Strand—consists of several bridge wires, of various diameters, twisted together.
- Galvanized Bridge Rope—consists of six strands twisted around a strand core.

Parallel Wire Cable

Tacoma Narrows Bridge Disaster

They Make Mistakes Too!
- Even good design cannot guarantee a perfect product
- Need testing of all products including software

Interests
- Testing
- Apply it to different types of software (such as web)
- Student introductions and their interests!

Goals of the Course
- Discuss advanced software testing techniques
- Two parts of the course
  - Review testing fundamentals
  - State-of-the-art & emerging techniques
- What do I expect from students?
**MS and Ph.D. Qualifying**

- Is the course is valid for PhD qualifying coursework?
  - Yes (Software Engineering/Programming Languages)
- Is the course is valid for MS qualifying coursework?
  - Yes (Software Engineering/Programming Languages)
- Is the course is valid for MS comps?
  - Yes (Both Midterm and Final exams count towards the MS comps.)

**Assessment**

- 25% Mid-term Exam
- 25% Final Exam
- 50% Project
  - Project report
  - Project Presentation

**Exam Contents**

- Midterm
  - Everything discussed in class
- Final exam
  - Everything discussed/presented after midterm

**Testing: Our Experiences**

**When to Stop?**

- Test Case Generation
- Software to be tested
- Output
- Verification

**A Real Testing Example**

**A Real Testing Example**

- Test Cases:
  - {1,2,3}
  - {1,3,2}
  - {3,2,3}
  - {}
  - {-1,-2}
- SPECS:
  - Takes a list of numbers; returns a sorted list
- Output:
  - {1,2,3}
  - {1,3,2}
  - {3,2,3}
  - {}
  - {-1,-2}
- Just a list
- A sorted list
- Repeated entry
- Empty list
- Negative numbers

**Philosophy:**

- What are we trying to do?
Automated Testing

Software to be tested

Test Case

Output

Verification

Enough?

Test Coverage

Yes

No

Test Case Generation

Verification

Test Case

Software to be tested

Output

Covered?

Test Coverage

Evaluator

Verifier

Test Oracle

OR

Test Case

Generator

Test Coverage

Evaluator

Verification

Automated Testing

Software to be tested

Test Case

Output

Coverage Evaluator

Veriﬁer OR Test Oracle

Testing the New Version

Original Test Cases

Original Software

Modified Software

New Test Cases

Regression Testing

Original Test Cases

Original Software

Modified Software

New Test Cases

Discussion

- Different Software Types
  - Object-oriented
  - Component-based
  - Concurrent
  - Distributed
  - Graphical-user Interfaces
  - Web
- Different goals of testing
  - Usability
  - Security
  - Correctness
  - Performance …